Algoritmos y Estructuras de Datos III - 1er Recuperatorio

Fecha de examen: 01-JUL-2022

	Nº Orden	Apellido y nombre		L.U.	# hojas [1]
Notas:	Ej1	Ej2	Ej3	Ej4	Final

Aclaraciones: El parcial se aprueba con 2 (dos) ejercicios aprobados. Cada hoja debe estar numerada y debe tener el número de orden y L.U. El parcial dura 4 horas y es a libro abierto. Cada respuesta dada debe contar con su correspondiente **justificación** para poder ser considerada correcta.

1) Se arrojan simultáneamente n dados, cada uno con k caras numeradas de 1 a k. Queremos calcular todas las maneras posibles de conseguir la suma total $s \in \mathbb{N}$ con una sola tirada. Consideramos que los dados son **indistinguibles**, es decir que si $n = 3$ y $k = 4$, entonces existen 3 posibilidades que suman $s = 6$:

    1) Dos dados valen 1 y el otro 4
    2) Un dado vale 1, otro 2 y otro 3
    3) Todos los dados valen 2

    a) Definir en forma recursiva la función $f\colon \mathbb{N}^3 \to \mathbb{N}$ tal que $f(n, s, k)$ devuelve la cantidad de formas de conseguir la suma s con n dados de k caras.

    b) Demostrar que f tiene la propiedad de superposición de subproblemas.

    c) Definir un algoritmo *top-down* para calcular $f(n, s, k)$ indicando claramente las estructuras de datos utilizadas y la complejidad resultante.

    d) Escribir el (pseudo-)código del algoritmo top-down resultante.

    Aprobación: definir y justificar correctamente f, indicando cómo se computa $f(n, s, k)$ en tiempo $O(nk \min\{s, nk\})$.

2) Como parte de la resolución del Ejercicio 1 del segundo parcial, queremos determinar aquellas aristas $v \to w$ de un digrafo D tales que $w \to v$ no es arista de D. Para ello, podemos usar las siguientes ideas sobre las estructuras de datos vistas en clase.

    a) Describir un algoritmo lineal que, dado un multigrafo G representado con un conjunto de aristas, determine las aristas (v, w) que no están repetidas en G.

    b) Describir un algoritmo lineal que, dado un digrafo D representado con un conjunto de aristas, determine las aristas $v \to w$ tales que $w \to v$ no es arista de D.

    Aprobación: justificar claramente las ideas de ambos algoritmos, que deben tomar tiempo lineal en el tamaño de la entrada.

3) La nueva reglamentación de una ciudad establece que tiene que haber una estación de policía a no más de 5 cuadras de cada esquina de la ciudad. Tenemos un grafo G cuyos n vértices representan las esquinas, donde sabemos cuáles esquinas están conectadas por cuadras. Queremos determinar todas las esquinas que no satisfacen la reglamentación, sabiendo que hay k estaciones de policía ubicadas en las esquinas $\{p_1, \ldots, p_k\}$. Diseñar un algoritmo que resuelva el problema en tiempo lineal. Explicar claramente su implementación y por qué es correcto.

    Aprobación: el algoritmo debe tomar tiempo lineal y se debe justificar correctamente el algoritmo.

[1] Incluyendo esta hoja.

4) Tenemos n casas unidas entre sí por m caminos, todos los cuales están tan rotos que no permiten el paso de los camiones que reparten soda. Se quieren reparar algunos de esos caminos de manera tal que todas las casas puedan ser visitadas por algún sodero. Conociendo las $r < n$ casas donde viven soderos y que el costo de reparar el camino que une las casas i y j es c_{ij} queremos determinar qué caminos debemos reparar para que todas las casas tengan acceso a un sodero de forma tal que la obra cueste lo menos posible. Afortunadamente, tenemos acceso la biblioteca FPGA (fast parallel graph algorithms), que permite resolver cada problema Π visto en la materia en tiempo $O(t(x)/\log t(x))$, donde $t(x)$ es el tiempo requerido por el mejor algoritmo visto en la materia para resolver Π sobre una instancia de tamaño x. Modelar el problema de los sodero para sacar el mayor provecho posible a la biblioteca FPGA, explicando claramente qué algoritmos de FPGA se utilizan y cuál es la complejidad resultante. Justificar por qué el modelo es correcto.

   Aprobación: el modelo debe ser correcto y estar bien explicado.